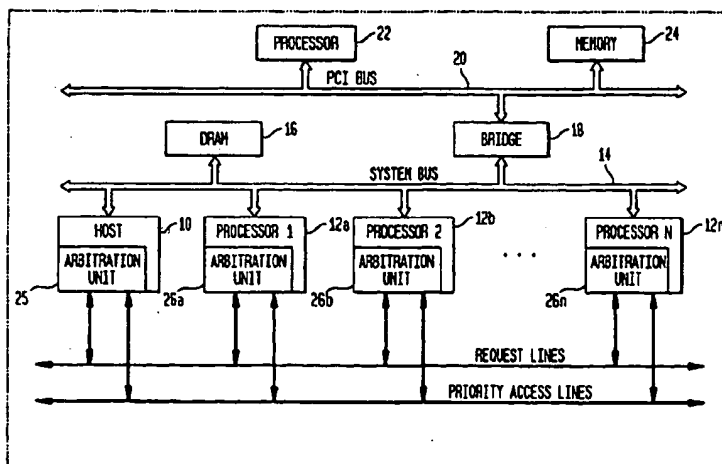




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : G06F 13/368, G11C 11/406	A1	(11) International Publication Number: WO 00/43895 (43) International Publication Date: 27 July 2000 (27.07.00)
(21) International Application Number: PCT/US00/00951 (22) International Filing Date: 14 January 2000 (14.01.00) (30) Priority Data: 09/235,780 22 January 1999 (22.01.99) US (71) Applicant: ANALOG DEVICES, INC. [US/US]; One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106 (US). (72) Inventors: KOSLAWSKY, Robert; Hachayot Street 4, 47214 Ramat-Hashron (IL). GREENFIELD, Zvi; Rivka Goobar Street 5, 44471 Kfar Sava (IL). SANDBANK, Alberto, E.; Wingate Street 8B, 30900 Zichron-Yaakov (IL). (74) Agent: PRITZKER, Randy, J.; Wolf, Greenfield & Sacks, P.C., 600 Atlantic Avenue, Boston, MA 02210 (US).		(81) Designated States: CN, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

(54) Title: DRAM REFRESH MONITORING AND CYCLE ACCURATE DISTRIBUTED BUS ARBITRATION IN A MULTI-PROCESSING ENVIRONMENT



(57) Abstract

A multiprocessor system includes a distributed bus arbitration system in which bus arbitration takes place simultaneously on each of the multiple processors connected to the bus. Each processor has a local arbitrator of common configuration with the other local arbitrators and a dedicated request line. Each local arbitrator is connected to each dedicated request line to monitor signals on lines indicative of requests for mastership of the bus by the processors. The multiprocessor system also includes a distributed DRAM refresh system in which each processor has a local DRAM refresh controller of common configuration with the other DRAM refresh controllers. Thus, as mastership of the bus passes from one processor to the other, the new bus master's local DRAM refresh controller can continue the DRAM refresh process without requiring information to be transferred from the old bus master to the new bus master, and without duplicating DRAM refresh operations.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

**DRAM REFRESH MONITORING AND CYCLE ACCURATE DISTRIBUTED BUS
ARBITRATION IN A MULTI-PROCESSING ENVIRONMENT**

Background of the Invention

Field of The Invention

5 The present invention relates generally to a multi-processor computer system and, more particularly, to a method and apparatus for distributing housekeeping functions between multiple processors in a multi-processor computer system.

Related Art

A computer typically has one or more central processing units ("processors") in
10 addition to dedicated functional units, such as a memory, a bus arbitration unit, a memory refresh controller, and one or more data and instruction caches, all of which will be referred to herein as "components." A computer system having more than one processor will be called a multi-processor system.

Communication between the processors and components takes place over a system
15 bus that interconnects the components and processors. Communication over the bus takes place in a "transaction." As is well known, instances may arise in which more than one processor is trying to issue a transaction on the bus at the same time. The process of determining which of the processors will control the bus and hence be allowed to issue a transaction on the bus in this situation is referred to as arbitration. The processor that wins
20 the arbitration is referred to herein as the "bus master."

Arbitration takes place in the bus arbitration unit. In a single processor system, the bus arbitration unit may be external to the processor or may be internal to the processor. A functional unit such as a bus arbitration unit will be referred to as "internal" when the functions performed by the functional unit are at least partially performed by the processor.
25 Conversely, a functional unit will be referred to as "external" when the functions performed by the functional unit are not performed by the processor. As is well known in the art, an external functional unit typically includes dedicated circuitry configured to perform a specific set of functions.

Figure 1 illustrates an exemplary multi-processor computer system using an external
30 bus arbitrator to perform bus arbitration. As shown in figure 1, the multi processor system has a host 10 and a plurality of processors 12a, 12b... 12n all connected to a system bus 14. An external arbitration unit 13 receives signals on bus request lines (R) and determines which

of the requesting processors should be granted bus mastership according to any known protocol. The processor (or host) that wins the arbitration is notified via a signal on an acknowledge line (A) connecting the arbitrator 13 and the processor or host. To become bus master, a processor asserts a signal on its bus request line (R) and waits for the external bus
5 arbitration unit 13 to grant it mastership of the bus via a signal on its associated acknowledge line (A). Arbitration using an external bus arbitrator thus requires both a request and an acknowledge line for each processor.

In many computer systems, including the system illustrated in Figure 1, different types of transactions are deemed to be more important than other types of transactions. For
10 example, a computer system may give transactions involving direct memory access (DMA) priority over other types of transactions. To accommodate requests of varying priority, it is common in such systems to interconnect the processors 12a-12n the and bus arbitration unit 13 with one or more priority request lines (P). When a processor desires to issue a priority transaction on the bus, the processor asserts a signal over its priority request line. The
15 external arbitration unit 13 processes the request and assigns the bus preferentially to the processor asserting the signal over the priority request line.

Arbitration using an external bus arbitration unit 13 may become space intensive since multiple dedicated lines must be provided to interconnect each processor 12a-12n and host 10 with the external arbitration unit 13. In the illustrated example, one request line, one
20 acknowledge line, and at least one priority request line are necessary to perform bus arbitration for each processor. When the number of processors connected to the bus is large, a significant amount of space is required to be used for arbitration.

In addition to multiple processors, computers typically employ solid-state memory components such as dynamic random access memory (DRAM) components. The term
25 DRAM will be used herein to refer to any type of dynamic random access memory, including DRAM (dynamic RAM), SDRAM (synchronous DRAM), DRDRAM (direct rambus DRAM), and other types of memory.

Digital information held in DRAM is generally in the form of charge stored on a two dimensional array of capacitors. The charge stored on each capacitor may be sensed during a
30 read operation or changed during a write operation. Additionally, charge is steadily lost over time due to leakage. The amount of leakage depends upon processing and operating conditions, as well as on environmental factors such as temperature and supply voltage.

Because of this leakage, the charge on individual capacitors in the capacitor array of the DRAM must be refreshed periodically.

To perform a refresh operation, digital information is conventionally read from the DRAM and immediately re-written into the DRAM. In this way, the charge on the individual
5 capacitors may be enhanced without changing the content of the memory array. To ensure the integrity of the content of the memory array, the DRAM must be refreshed before the leakage causes the charge stored in the capacitors to drop below a predetermined level. Since DRAM is typically organized in rows and columns, refresh of the DRAM is performed typically row-by-row, although other refresh schemes are also possible.

10 Control of the DRAM refresh procedure in a multiple processor environment is typically handled by an external DRAM refresh controller. The external DRAM refresh controller has a refresh counter indicating the row to be refreshed and a reset enabling the refresh counter to be reset, for example, to the first row. Although the functions performed by the DRAM refresh controller are important to operation of the computer, the DRAM
15 refresh controller consumes valuable space that otherwise could be used for other components. Additionally, the DRAM refresh controller must be connected to the DRAM either through one or more dedicated lines or over the system bus. When the DRAM refresh controller uses the system bus to control DRAM refresh operations, system performance as a whole is degraded by reducing the amount of data that otherwise could be transferred. When
20 the DRAM refresh controller is connected to the DRAM through one or more dedicated lines, valuable space is consumed that could otherwise be used for other purposes.

Accordingly, what is needed is a method and apparatus that will facilitate performance of maintenance functions, such as bus arbitration and DRAM refresh operations, in a multi-processor environment while minimizing the amount of space and other system
25 resources required to perform these functions.

Summary of the Invention

The present invention is a method and apparatus that will facilitate performance of maintenance functions, such as bus arbitration and DRAM refresh operations, while
30 minimizing the amount of space required to perform these functions in a multi-processor environment.

In one embodiment of this invention, each processor is provided with a local arbitration unit of common configuration with the other local arbitration units to perform bus

arbitration simultaneously. Each local arbitration unit monitors requests to become bus master by itself and other processors, is synchronized with the other arbitrators, and is provided with similar logic. Accordingly, each local arbitration unit independently determines the identity of the subsequent bus master. By providing local arbitrators that each
5 independently determine allocation of bus mastership, the multi-processor system is able to eliminate the need for acknowledge lines associated with arbitration. This greatly reduces the amount of area required for arbitration.

Similarly, to perform DRAM refresh operations, each processor is provided with a local DRAM refresh controller having at least one pin connected to one or more dedicated
10 DRAM refresh lines. In a preferred embodiment, each processor has one pin connected to a single DRAM refresh line. Each local DRAM refresh controller, when responsible for performing DRAM refresh operations, issues DRAM refresh signals to the DRAM over the DRAM refresh line. When not responsible for performing DRAM refresh operations, each local DRAM refresh controller monitors the DRAM refresh line, senses DRAM refresh
15 signals issued by other DRAM refresh controllers, and uses the sensed signals to update its DRAM refresh row counter. A reset line may also be provided to reset the counter. Alternatively, reset of the controller may be performed via the single DRAM refresh line. Providing each processor with a local DRAM refresh controller eliminates the need for an external DRAM refresh controller thus saving space and removing the DRAM refresh
20 controller as a contending component for mastership of the bus. Also, by ensuring that each DRAM refresh controller has current information relating to DRAM refresh operations, duplication of refresh operations is minimized during hand-offs between processors. Thus, the newly responsible DRAM refresh controller is able to continue the DRAM refresh operation where the last processor's DRAM refresh controller stopped.

25

Brief Description Of The Drawings

This invention is pointed out with particularity in the appended claims. The above and further advantages of this invention may be better understood by referring to the following description when taken in conjunction with the accompanying drawings, in which:

30

Figure 1 is a functional block diagram of a conventional multi processor system;

Figure 2 is a functional block diagram of a multi processor system according to one embodiment of this invention;

Figure 3 is a functional block diagram of a bus arbitration system for use in the multi-processor system of figure 2;

Figure 4 is a functional block diagram of a processor for use in the bus arbitration system of figure 3; and

5 Figures 5-8 are timing diagrams illustrating aspects of the bus arbitration protocol according to one embodiment of this invention.

Detailed Description

In one embodiment, the multi processor system of this invention includes a distributed
10 bus arbitration system in which bus arbitration takes place simultaneously on each of the multiple processors connected to the bus. Each processor has a local arbitrator of common configuration with the other local arbitrators and a dedicated request line. Each local arbitrator is connected to each dedicated request line to monitor signals on lines indicative of requests for mastership of the bus by the processors. Since each local arbitrator is of
15 common configuration with the other local arbitrators, is operating synchronously with the other arbitrators, and is provided with a similar set of inputs, each arbitrator will arrive at the same conclusion as to which processor is to become bus master. Accordingly, an external bus arbitrator is not required and acknowledge lines are not required to communicate signals indicative of the result of the bus arbitration to the processors. Additionally, the number of
20 priority request lines can be dramatically reduced by requiring a requesting processor to de-assert a bus request upon detecting a priority request by another processor.

In another embodiment, the multi processor system includes a distributed DRAM refresh system in which each processor has a functionally equivalent local DRAM refresh controller. Signals on a refresh line are monitored by each of the local DRAM refresh
25 controllers to enable synchronous updating of local DRAM refresh counters and resets of the local DRAM refresh counters. Thus, as mastership of the bus passes from one processor to the other, the new bus master's local DRAM refresh controller can continue the DRAM refresh process without requiring information to be transferred from the old bus master to the new bus master, and without duplicating DRAM refresh operations.

30 In the multi processing system of figure 2, a host 10 and a plurality of processors 12a-12n are connected by a system bus 14. A DRAM 16 is connected to the system bus 14 and a bridge 18 connects the system bus to an external bus 20, such as a peripheral component interconnect (PCI) bus. Components such as a processor 22 and a memory 24 may be

connected to the PCI bus 20, as may numerous other components. Each processor 12a-12n includes a local arbitration unit 26, 26a-26n interconnected by request lines and priority access lines. The host 10 is provided with a host arbitration unit that may be of common configuration with the local arbitration units 26a-26n or may be differently configured, as
5 described in more detail below. The interconnection between the system bus 14, host 10 and processors 12, as well as the arbitration lines are illustrated in more detail in figure 3.

As shown in figure 3, a distributed arbitration system includes a host 10 and processors 12a-12n connected to the system bus 14. The system bus 14 may be any known bus over which transactions may take place. Local arbitrators 26a-26n are contained in each
10 processor 12a-12n. A host arbitrator 25 is similarly contained in the host 10. Although arbitrators 25, 26a-26n may be external to the host 10 and processors 12a-12n, it is currently envisioned that the arbitrators 25, 26a-26n will be internal to the host 10 and processors 12a-12n respectively. The local arbitrators 25, 26a-26n may be formed of common configuration. Alternatively, the host arbitrator 25 may be configured differently than the local arbitrator
15 26a-26n. Forming the local arbitrators 26a-26n and optionally the host arbitrator 25 of common configuration enables the arbitrators to each arrive at the same result given a particular set of inputs.

In the following discussion, it is assumed that the host 10 has a host arbitration unit 25 that is not of common configuration with the local arbitration units 26a-26n. If, however, it
20 was desired to do so, the host 10 could be provided with a commonly configured local arbitration unit. In this situation, arbitration with the host 10 could take place in a manner similar to the remaining processors. Figure 9 illustrates this alternative embodiment. When one of the processors needs to issue a transaction on the system bus 14, the processor will assert a signal to become bus master on its associated bus request line. Since every local
25 arbitrator 26a-26n is connected to the processor's bus request line, every local arbitrator 26a-26n will receive the signal on processor's bus request line and use that input in any upcoming arbitration. Since all local arbitrators 26a-26n in a cluster of processors 12a-12n operate synchronously (in lock-step on a cycle by cycle basis), every processor 12a-12n keeps track and follows the arbitration sequence of the system bus 14. If, therefore, the result of a
30 subsequent arbitration is that the requesting processor is to become bus master, mastership will be granted to the requesting processor without requiring the arbitration result to be communicated to the requesting processor. In this instance, all other processors will

recognize the requesting processor's right to master the system bus 14 and refrain from issuing transactions on the system bus 14. Likewise, if the result of the arbitration is that the requesting processor is not to become bus master, its associated local arbitrator will prevent the requesting processor from attempting to issue a transaction on the system bus 14.

5 One unique aspect of using local arbitrators 26a-26n instead of a remote arbitrator 13 is that there is no need to communicate the result of the arbitration to the processors 12a-12n. Instead, each processor 12a-12n is provided with the arbitration result directly from its local arbitrator 26a-26n. Accordingly, acknowledge lines are not required. This substantially reduces the amount of space required to perform arbitration.

10 As shown in figures 3 and 4, bus arbitration is accomplished using signals on lines BR0-BR7, HBR and HBG. The signals may be digital, analog, optical or any other type of signal useable to communicate information between host 10 and processors 12a-12n. Signals on lines BR0-BR7 are used to arbitrate between processors 12a-12n, and signals on lines HBR and HBG are used to pass control of the system bus 14 to and from the host 10. In this
15 context, a signal on line BR0 is used by processor 0 to request mastership of the bus 14, a signal on BR1 is used by processor 1 to request mastership of the bus 14, etc. A signal on line HBR is used by the host 10 to request mastership of the bus 14 and a signal on line HBG is used to pass mastership of the bus 14 to the host 10. Dedicated pins on the host 10 and processors 12a-12n enable continuous access to and monitoring of signals on the lines used
20 for arbitration.

Since the local arbitrators 26a-26n must have a dedicated input pin to monitor signals on the request line of the host 10 and each processor 12a-12n taking place in the arbitration, the number of processors 12a-12n that may arbitrate for mastership of the system bus 14 is limited by the hardware used to implement the local arbitrators 26, 26a-26n. Accordingly,
25 for example in the illustrated embodiment, each arbitrator 26, 26a-26n has 10 input pins connected to lines HBR, HBG and BR0-BR7. Thus, in this embodiment, up to 8 processors 12a-12n can arbitrate for mastership of the system bus 14 simultaneously. If fewer than the maximum number of processors are connected to the system bus 14, the unused pins are de-asserted to prevent the local arbitrators from attempting to grant bus mastership to a non-
30 participating processor. Although the illustrated embodiment shows eight processors connected to the system bus 14, this number may be varied by altering the number of bus request lines and pins or by de-asserting several of the bus request lines. Accordingly, the

concepts disclosed herein are not limited to a system using eight processors, and apply equally to a system using at least two processors.

Each processor 12a-12n is provided with an identification number. In an exemplary embodiment, the identification number is assigned to the processor 12a-12n using one or
5 more identification pins 28a-28n. Any other method of identifying the processors may be used as well.

When one of the processors 12x desires to become master of the bus, it asserts signal on its own bus request line, monitoring at the same time the other bus request lines for signals. If no signal from a processor with a higher priority is asserted on another bus request
10 line, mastership of the bus is passed to the new requester after the current bus master completes its transactions. The new bus master retains bus mastership by asserting the signal on its bus request line continuously as long as it is the master. The processor that is bus master retains mastership of the bus until (1) it has no more pending transactions; (2) a counter 30 expires and mastership is passed to another requesting processor to prevent
15 starvation of the bus; or (3) its mastership is interrupted to enable higher priority transactions to take place. Transfer of mastership occurs by the current bus master de-asserting its bus request line.

When the current master relinquishes the system bus 14, it de-asserts the signal on its associated bus request line, and is assigned the lowest priority. In the present embodiment,
20 an idle cycle is inserted when mastership is passed from the current master to the new master. Pipelining mastership of the bus may be possible using known techniques to eliminate this idle cycle, although in the presently preferred exemplary embodiment there is no pipelining between masters.

A rotating bus priority scheme is implemented to ensure bus fairness between
25 processors 12a-12n. If two or more processors simultaneously assert bus request signals on their associated bus request lines the processor with the highest priority is granted mastership of the bus. Priority between processors may be determined in any other known manner.

When a requesting processor is not granted mastership of the bus, the requesting processor continuously asserts its bus request signal on its associated bus request line until it
30 is granted mastership of the bus, it is required to de-assert the signal to accommodate a higher priority request or it no longer requires mastership of the system bus 14.

It may be desirable to limit the time that a single processor retains mastership of the bus to prevent bus starvation of other processors in the system. To provide this limit, the counter 30 (see figure 4) is provided in each local arbitrator 26a-26n. The counter 30 defines the maximum amount of time that the processor may master the bus 14 when other
5 processors are also requesting mastership of the bus. The counter 30 is initiated to a predefined value when the processor attains mastership of the system bus 14 and starts counting up or down. Counting is paused if mastership of the bus is temporarily transferred to the host or another processor with a higher priority transaction. If the counter 30 reaches a second predefined value, and another processor is requesting mastership of the bus 14, the
10 processor completes its transaction and relinquishes mastership of the system bus 14 for at least one cycle. If no other request for mastership of the system bus 14 has been asserted when the counter 30 reaches the second predefined value, the counter 30 is reset and starts to count up or down once again while the processor 12 continues to retain mastership of the bus. In one preferred embodiment, one of the first and second predefined values is zero, thus
15 resulting in a system with only a single predefined value.

As alluded to above, several types of transactions are deemed to be more important than other types of transactions. To accommodate these transactions, a processors or the host
10 with a priority transaction can temporarily interrupt the current bus master and obtain temporary mastership of the system bus 14. In the illustrated embodiment, a processor with
20 core priority access (CPA) transaction or a DMA priority access (DPA) transaction is allowed to interrupt the current bus master and obtain temporary mastership of the bus. To facilitate these higher priority transactions, a CPA line and a DPA line are provided over which processors 12a-12n may assert priority request signals to obtain mastership of the bus. Likewise, the host 10 can obtain temporary mastership of the bus by asserting a signal on a
25 host bus request line HBR. Use of the CPA, DPA and HBR lines is discussed in greater detail below.

A processor 12x will assert a signal on the CPA priority request line when it has a core transaction that needs to accesses external memory. This allows a processor to interrupt background transfers of a DMA channel belonging to the bus master and gain control of the
30 system bus 14. The current bus master in this situation terminates its transaction and passes mastership of the bus to the requesting processor by de-asserting the signal on its associated bus request line. When a signal is asserted on the CPA priority request line, only processors

with core transactions can request the bus 14. The other requesting processors de-assert their bus request lines when they sense that a signal on the CPA priority request line has been asserted. When more than one processor requests the system bus 14 by asserting a signal on its bus request line along with a signal on the CPA priority request line, the processor with
5 the highest priority gains bus mastership. Allocation of bus mastership between processors of equal priority is discussed above, and may be implemented using any known protocol.

Requiring those processors not asserting a signal on the CPA priority request line to de-assert signals on their respective bus request lines enables priority requests to be processed simultaneously by the local arbitrators 26a-26n using a single line. Specifically, since the
10 protocol requires all processors not asserting a signal on the CPA priority request line to de-assert signals indicative of requests for mastership of the bus once a signal on the CPA priority request line is sensed, any bus request line that is asserted when a signal on the CPA priority request line is asserted necessarily must have a high priority transaction.

Accordingly, arbitration between multiple processors with simultaneous priority transactions
15 can take place using only a single priority line. An example may be illustrative of this process.

With reference to figure 3, suppose that the following signals were asserted simultaneously:

20 Processor 1 — BR1, CPA
Processor 2 — BR2, CPA
Processor 3 — BR3
Processor 4 — BR4.

Upon determination by processors 3 and 4 that a signal on the CPA priority request line had
25 been asserted, processors 3 and 4 de-assert signals on their bus request lines BR3 and BR4 respectively. At this point in time, only signals on the bus request lines associated with processor 1 and 2, (BR1, BR2) and a signal on the CPA priority request line are asserted. Accordingly, the arbitrators 26a-26n realize that two processors have simultaneously asserted signals on the CPA priority request line, and thus that two high priority requests have been
30 made. Accordingly, the arbitrators 26a-26n can allocate priority of bus mastership between processors 1 and 2 according to an ordinary priority scheme. In this manner, interpreting combinations of asserted lines enables priority access to the system bus 14 to be handled using a single priority request line without requiring a dedicated priority request line for each processor 12.

Multiple levels of priority can be implemented in a similar fashion using additional priority access lines. For example, in the embodiment illustrated in figure 3, a second level of priority is provided using a DPA priority request line. A signal on the DPA priority request line is asserted by a processor when the processor needs to perform a high priority DMA transaction involving external memory. Since these transactions are deemed to be of lower priority than CPA transactions, a processor 12a-12n will only assert a signal on the DPA priority request line when a signal on the CPA priority request line is not asserted. Assertion of a signal on the DPA priority request line allows a high priority DMA channel belonging to a slave processor 12 to interrupt background transfers of a regular priority DMA channel belonging to a master processor and gain temporary mastership of the bus 14.

When a signal on the DPA priority request line is asserted, the current master terminates its transaction and passes bus mastership to the requesting processor by de-asserting the signal on its bus request line. Likewise, any other requesting processors that do not have a transaction for which assertion of a signal on the DPA priority request line or CPA priority request line would be appropriate, de-assert signals on their bus request lines when they sense that a signal on the DPA priority request line is asserted. Stated differently, when a signal on the DPA priority request line is asserted, only processors 12a-12n with core transactions or transactions involving high priority DMA transfers can request the bus.

When more than one processor requests the system bus 14 by asserting a signal on its respective bus request line along with a signal on the DPA priority request line, the processor with the highest priority gains bus mastership. By requiring processors 12a-12n to de-assert signals on their respective bus request lines when a signal on the DPA priority request line is asserted (and they are not asserting a signal on the DPA priority request line) a single line may be used to provide DMA priority access to all processors 12a-12n. Using a combination of signals on the bus request lines and priority request lines enables the bus arbitration units to accommodate multiple priority levels using a minimum number of lines.

In this arbitration process, requesting processors asserting a signal on the CPA priority request line have higher priority over requesting processors asserting signals on the DPA priority request line. Thus, requesting processors asserting a signal on the DPA priority request line will de-assert signals on their respective bus request lines (and the DPA priority request line) upon sensing that a signal on the CPA priority request line has been asserted by another processor.

Other priority schemes may be used as well, involving more or fewer priority request lines. For example, three, four or more priority request lines could be used to provide for three, four or more levels of priority. Also, simultaneous assertion of signals on two or more priority request lines (such as simultaneous assertion of signals on the CPA priority request line and DPA priority request line) by the same processor may be used to provide additional levels of priority. Accordingly, combinations of request signals and priority signals enable multiple levels of priority to be accommodated using a minimum number of priority request lines.

The host 10 can be the bus master. When the host 10 would like to become bus master, it causes its host arbitration unit 25 to assert a signal on a host bus request line (HBR). The current bus master, upon determining that a signal has been asserted on line HBR, completes its current transaction and relinquishes the system bus 14 by asserting a signal on a host bus grant line (HBG). The host 10, maintains the signal on line HBR asserted as long as it requires the bus. The master which last relinquished the system bus 14 maintains a signal on its respective bus request line asserted so that when the host 10 de-asserts the signal on line HBR, it retains mastership of the bus.

Occasionally, the bus master may engage in transactions that cannot safely be interrupted. Accordingly, it may become important for the bus master to lock the system bus 14 to prevent mastership of the bus from being temporarily reassigned. Accordingly, the arbitration system provides the ability for a processor 12 to lock the bus. Specifically, to lock the bus, the bus master asserts and maintains a signal on its bus request line. obtains mastership of the bus, and then asserts a signal on a line dedicated to locking the bus, referred to herein as the lock line. When a signal on the lock line is asserted, assertion of a signal on the CPA or DPA priority request lines or the host bus request lines will not affect mastership of the bus. Similarly, expiration of the counter 30 will not cause mastership of the system bus 14 to be relinquished. The only situation where mastership of the system bus 14 is affected after the system bus 14 has been locked is when a signals on both the host bus request line and the backoff line are asserted simultaneously. After the bus master has completed its transaction requiring the system bus 14 to be locked, the signal on the lock line is de-asserted and ordinary arbitration resumes.

Each processor is also provided with a back-off pin connected to a line used to back the current processor off the bus, referred to herein as the backoff line. Unless a signal on the

lock line has been asserted, the processor that is the current bus master will immediately cede mastership of the bus 14 upon assertion of a signal on the backoff line. This process is described below.

Referring now to figure 2, a situation may occur where a local processor would like to
5 issue a transaction to read from the memory attached to the PCI bus 20. The local processor will arbitrate for and obtain mastership of the system bus 14 and issue the transaction to the PCI bus 20 over the bridge 18. At the same time, however, the processor 22 on the PCI bus 20 may be attempting to read from a memory connected to the system bus 14 or from the local processor's memory. In this situation, both the local processor and the external
10 processor 22 connected to the PCI bus 20 are attempting to gain access to the system bus 14 simultaneously. Also, since the local and external processors are not associated with each other, there is no arbitration protocol that will determine mastership between the local processor and the external processor 22. This situation is known as deadlock.

To resolve deadlock, each processor is provided with a backoff pin connected to the
15 backoff line. The bridge 18, sensing that deadlock has occurred, will assert a signal on the backoff line. Assertion of a signal on the backoff line terminates mastership of the system bus 14 to all processors 12 and the host 10 pending completion of the transaction by the remote processor 22. Upon completion of the transaction by the remote processor 22, the bridge 18 will de-assert the signal on the backoff line, and the local processor 12 that
20 originally had mastership of the system bus 14 will be allowed to complete its transaction. In this case, the local processor with mastership of the system bus 14 would be allowed once again to attempt to access the memory connected to the PCI bus 20.

One embodiment of an arbitration protocol will now be described in more detail with reference to the signal diagrams illustrated in figures 5-8. Any arbitration protocol capable of
25 allocating mastership of the system bus 14 between the host 10 and processors 12a-12n could be used instead of or in addition to the illustrated protocol. The exemplary protocol illustrates one way of passing mastership of the bus between the multiple processors and host equitably, while enabling higher priority tasks to be granted preferential access to the bus. In the described protocol, mastership of the bus is granted to a requesting device in the
30 following order of priority:

- (1) Host 10 (a signal on HBR asserted)
- (2) Processor 12 (a signal on BR along with a signal on CPA asserted)
- (3) Processor 12 (a signal on BR along with a signal on DPA asserted)

(4) Processor 12 (a signal on BR asserted).

Even if the processor has mastership of the bus, however, there are several instances where the processor will be required to yield mastership temporarily to another requesting processor or host 10. These conditions include:

- 5 (1) A signal on the backoff line is asserted and signals on the host bus grant and lock lines are not asserted (external processor obtains temporary mastership of bus 14).
- (2) The current master does not request the system bus 14 and another processor or the host 10 asserts its bus request line or host bus request line (new processor or host 10 obtains mastership of bus 14).
- 10 (3) The current master requests the system bus 14 by asserting a signal on its bus request line and optionally a signal on the DPA priority request line but not the CPA priority request line, and another processor asserts a signal on its bus request line and the CPA priority request line (new processor becomes temporary bus master).
- 15 (4) The current master requests the system bus 14 by asserting a signal on its bus request line but not on the CPA or DPA priority request lines, and another processor asserts a signal on its bus request line and the DPA priority request line (new processor becomes temporary bus master).
- 20 (5) The current master request the system bus 14 by asserting a signal on its bus request line and optionally a signal on the CPA or DPA priority request line, and the host 10 asserts a signal on the host bus request line (host 10 becomes temporary bus master upon completion of the current transaction and assertion of a signal on the host bus grant line by the current master).
- 25 (6) The current master request the system bus 14 by asserting a signal on its bus request line and optionally a signal on the CPA or DPA priority request line, but the value of the counter 30 expires and another processor asserts a signal on its bus request line (new processor becomes bus master).

Signal diagrams of signals on arbitration lines are illustrated in figures 5-8. In these
 30 figures, arbitration takes place as discussed above in multiple commonly configured local arbitration units 26a-26n simultaneously. All arbitration units are operating synchronously. A clock signal, MCLK, is provided to time operations of the arbitration units and the system bus. To facilitate understanding of the arbitration process, only signals on bus request lines associated with three processors (BR0, BR1 and BR2) are illustrated. In the signal diagrams
 35 of figures 5-8, a signal is asserted when the signal is low. The converse convention could be adapted as well.

Figure 5 illustrates the arbitration protocol between multiple requesting processors in the absence of any priority requests. In figure 5, during clock cycle 1, processor 0 is the bus master, a signal on bus request line BR0 is asserted and signals on bus request lines BR1, and
 40 BR2 are not asserted. Signals on priority request line CPA, DPA and the host bus request

line HBR are also not asserted in this scenario, and will remain not asserted during the entire arbitration illustrated in figure 5. Assertion of signals on line CPA, DPA, HBR and HBG will be discussed more fully below in connection with figures 6-8.

During clock cycle 2, a signal on line BR1 is asserted, and during clock cycle 3, a
5 signal on line BR2 is asserted. Since assertion of a signal on lines BR1 or BR2 cannot disrupt the current bus master's right to continue as bus master, assertion of signals on lines BR1 and BR2 have no immediate effect on mastership of the bus. Accordingly, processor 0 remains bus master until it completes its transactions and de-asserts the signal on its bus request line BR0 at cycle 5. Cycle 5 is a turn over cycle during which no processor is master
10 of the system bus 14 and no data is transferred over the bus. At cycle 6, processor 1 becomes bus master due to its priority over processor 2. Note that the order in which the processors request the system bus 14 is not determinative of which processor will become the next bus master. Instead, as discussed above, to assure equitable access to the system bus 14 by all processors, a rotating priority scheme is used. Although a rotating priority scheme is used in
15 this embodiment, other priority schemes could be used as well. Thus, processor 1 would become bus master even if the order in which processors 1 and 2 requested the system bus 14 was reversed.

After becoming bus master, processor 1 remains bus master until it has completed its transaction at cycle 10. While processor 1 is bus master, processor 0 asserts a signal on line
20 BR0 at cycle 9. After the turn over cycle (cycle 11), mastership of the system bus 14 transfers to processor 2. Transfer of mastership to processor 2 occurs because processor 2 has priority over processor 0 (processor 0 has the lowest priority due to the fact that it was the last processor to relinquish mastership of the bus 14). Upon completion of its transaction at cycle 16, processor 2 cedes mastership of the system bus 14 to processor 0, which assumes
25 mastership of the system bus 14 during cycle 17.

Mastership of the bus may be interrupted either when the host 10 (having the highest priority in the system) requests the system bus 14 by asserting a signal on the host bus request line HBR or when one of the other processors requests the system bus 14 and asserts a signal on one of the priority request lines CPA or DPA. Arbitration in the presence of a signal on a
30 priority request line will now be discussed with reference to figures 6-8.

Figure 6 is a signal diagram illustrating the arbitration protocol when a signal on the CPA priority request line is asserted. As shown in figure 6, processor 0 is initially the bus

master. During cycle 1, processor 1 asserts a signal on line BR1. As noted above, however, this does not affect bus mastership. Thus, during cycle 2, processor 0 retains bus mastership.

During cycle 3, processor 2 asserts signals on lines BR2 and CPA. Since assertion of a signal on line CPA does affect temporary bus mastership, assertion of signals on lines BR2 and CPA causes processor 1 and processor 0 to take action. Specifically, processor 1, during the next clock cycle (cycle 4) de-asserts the signal on line BR1. Likewise, during cycle 5, processor 0 de-asserts the signal on line BR0 to cede temporary mastership of the system bus 14 to processor 2. After processor 2 has completed its priority transactions at cycle 10, it cedes control of the system bus 14 (cycle 11) by de-asserting the signal on its bus request line and the signal on line CPA. During cycle 12, both processor 1 and processor 0 assert signals on their respective bus request line. However, since processor 0 had mastership of the system bus 14 before the assertion of signals on lines CPA and BR2 by processor 2, processor 0 has priority over processor 1. Accordingly, processor 0 regains mastership of the system bus 14 and is able to complete its transactions. Upon completion of its transactions, processor 0 relinquishes mastership of the system bus 14 at cycle 17 and processor 1 becomes bus master in cycle 18.

Figure 7 is a signal diagram of an arbitration involving multiple processors, one of which asserts a signal on the DPA priority request line. As shown in figure 6, processor 0 initially has mastership of the bus. During cycle 1, processor 1 asserts a signal on line BR1 without asserting a signal on line CPA or line DPA. As discussed above with respect to figure 5, this does not affect mastership of the bus. Accordingly, processor 0 continues to be master of the bus.

During cycle 3, processor 2 asserts signals on both line BR2 and the DPA priority request line. Since this does affect mastership of the bus, during cycle 4, processor 1 de-asserts the signal on line BR1 and processor 0 de-asserts the signal on line BR0. Accordingly, after an idle turn-over cycle (cycle 5), processor 2 becomes bus master.

Upon completion of its transaction, processor 2 de-asserts the signals on lines BR2 and DPA, thus relinquishing mastership of the bus. This causes processor 1 to reassert a signal on line BR1 during cycle 11 and causes processor 0 to reassert a signal on line BR0 during cycle 12. Since processor 0 was the master prior to assertion of the signal on line DPA, processor 0 has priority over processor 1 even though processor 1 reasserted a signal on its bus request line one cycle earlier than processor 0 reasserted a signal on its bus request

line. Thus, at cycle 13, processor 0 regains mastership of the bus. Upon completion of its transaction, processor 0 de-asserts the signal on line BR0 at cycle 17, thus relinquishing the system bus 14 to the processor with the then highest priority. As shown in figure 6, in this instance, the processor with the next highest priority is processor 1. Processor 1 will retain
5 mastership of the system bus 14 by asserting a signal on line BR1 during its transaction.

In the arbitration process, a processor asserting a signal on the CPA priority request line has a higher priority over a processor asserting a signal on the DPA priority request line. Thus, a processor asserting a signal on line DPA will de-assert the signal on line DPA and its associated bus request line upon sensing that another processor has asserted a signal on line
10 CPA. Since this is straightforward, no further explanation will be provided.

Figure 8 is a signal diagram of an arbitration involving the simultaneous assertion of two signals on line CPA and assertion by the host 10 of a signal on line HBR. In figure 8, during cycle 1, processor 0 is the master and a signal on line BR0 is asserted. During cycle 2, processors 1 and 2 both assert signals on their respective bus request lines, BR1 and BR2,
15 while also simultaneously asserting signals on line CPA. During cycle 3, the host 10 asserts a signal on line HBR.

In response, processor 0 completes its transaction. As shown in this figure, completion of processor 0's transaction may take a cycle or more. Accordingly, in cycle 5, after completion of processor 0's transaction, processor 0 de-asserts the signal on line BR0
20 and simultaneously asserts a signal on line HBG.

Assertion of a signal on line HBG passes mastership of the system bus 14 to the host 10. The host retains mastership of the system bus 14 for as long as necessary by asserting and maintaining a signal on line HBR. Upon completion of its transaction, the host 10 de-asserts the signal on line HBR at cycle 10 to cede mastership of the bus and to return
25 mastership of the bus to processor 0. However, since processors 1 and 2 have both asserted signals on lines BR1 and BR2 and signals on line CPA, mastership does not return to processor 0 as would otherwise be the case. Instead, since assertion of a signal on line CPA by one or more processors takes priority over a non-priority transaction, mastership passes to one of the processors asserting a signal on line CPA. In this instance, since the signals on
30 line CPA have been asserted by both processor 1 and processor 2, the processor with higher priority is granted mastership of the bus. In the illustrated example, processor 1 is given priority over processor 2 and mastership passes to processor 1 at cycle 12. Upon completion

of its transaction, processor 1 relinquishes mastership of the system bus 14 by de-assertion of the signal on its bus request line BR1 and de-assertion of the signal on line CPA (by processor 1) to pass mastership of the system bus 14 to processor 2 at cycle 16.

Note, in this instance, assertion of a signal on line HBR enabled processor 0 to
5 complete its transaction even though a signal on CPA had been asserted by another processor. If this were not the case and processor 0 had not been allowed to complete its transaction prior to relinquishing bus mastership to the host 10, upon completion of the transaction of processor 2, mastership of the system bus 14 would return to processor 0 — i.e., processor 0 would have priority over other processors subsequently asserting signals on their respective
10 bus request line without simultaneously asserting a signal on line CPA or DPA.

In addition to a local arbitrator, each processor is also provided with a local DRAM refresh controller 32. As discussed above, DRAM must be refreshed periodically and typically is refreshed by row. As such, a DRAM refresh controller 32 typically has a row counter 34 indicating the row of DRAM last refreshed and a reset for resetting the row
15 counter 34 when the last row of DRAM is refreshed.

Handing off the DRAM refresh between multiple processors 12 is facilitated by providing each processor with an local DRAM refresh controller 32. Additionally, as shown in figure 9, the host 10 may be provided with a DRAM refresh controller 32. Providing the host with a DRAM refresh controller 32 is only necessary when the host can itself access the
20 DRAM. Each local DRAM refresh controller 32 is provided with a refresh input pin to receive signals on a refresh line. Signals related to the refresh process are asserted over the line so that all local DRAM refresh controllers 32 have the same input. These signals can be used to increase the DRAM refresh counter 34 or to reset the counter 34. Optionally, the DRAM refresh counter 34 can be configured to automatically revert to the first predefined
25 value upon reaching the second predefined value. Because all local DRAM refresh controllers 32 are of common configuration, are synchronized and are provided with similar inputs, all local DRAM refresh controllers 32 are current as to the state of the DRAM refresh process. Accordingly, when mastership of the system bus 14 is transferred from one processor 12 to another processor 12, the local DRAM refresh controller 32 is able to
30 continue the DRAM refresh process where the prior bus master stopped without excessive duplication.

When mastership of the bus passes to the host 10, the refresh controller makes the DRAM enter a self-refresh state before the bus is granted to the host. In this situation, the host does not need to conduct any refresh operation, and indeed is not provided with a DRAM refresh controller (Figure 3). Optionally, if it was desired to allow the host 10 to
5 access the DRAM directly, the host 10 may be provided with a DRAM refresh controller of common configuration with the other DRAM refresh controllers contained in the processors 26a-26n as shown in figure 9.

It should be understood that various changes and modifications of the embodiments shown in the drawings and described in the specification may be made within the spirit and
10 scope of the present invention. Accordingly, it is intended that all matter contained in the above description and shown in the accompanying drawings be interpreted in an illustrative and not in a limiting sense. The invention is limited only as defined in the following claims and the equivalents thereto.

What is claimed is:

CLAIMS

1. An apparatus, comprising:
 - a bus;
 - 5 a first processor connected to the bus;
 - a second processor connected to the bus;
 - a first local arbitrator associated with the first processor and configured to arbitrate mastership of the bus between the first and second processors; and
 - a second local arbitrator associated with the second processor and configured to
 - 10 arbitrate mastership of the bus between the first and second processors.
2. The apparatus of claim 1, wherein said second local arbitrator is of common configuration to the first local arbitrator.
- 15 3. The apparatus of claim 2, further comprising:
 - a first bus arbitration line connected to the first and second local arbitrators and associated with the first processor over which the first processor can assert a signal indicating a request for mastership of the bus; and
 - a second bus arbitration line connected to the first and second local arbitrators and
 - 20 associated with the second processor over which the second processor can assert a signal indicating a request for mastership of the bus.
4. The apparatus of claim 3, wherein the first arbitrator is internal to the first local processor; and wherein the second local arbitrator is internal to the second processor.
- 25 5. The apparatus of claim 1, further comprising multiple additional processors and multiple additional local arbitrators, all interconnected by bus arbitration lines.
6. The apparatus of claim 1, further comprising:
 - 30 a host,
 - a local host arbitrator associated with the host and configured to arbitrate mastership of the bus between the host, first processor and second processor.

7. The apparatus of claim 3. further comprising:

at least one priority access line connected to the first and second local arbitrators and not associated with the first or second processor, wherein assertion of a signal on the priority access line results in at least one of a higher priority to the processor asserting the signal on the priority access line in an arbitration between the first and second processors, and a temporary disruption in mastership of the bus by the processor asserting the signal on the priority access line over a processor not asserting the signal on the priority access line.

8. The apparatus of claim 3. further comprising:

10 a backoff line connected to the first and second local arbitrators for remediating a deadlock condition on the bus.

9. An apparatus, comprising:

a bus;

15 a first processor connected to the bus;

a second processor connected to the bus;

a memory,

a first local memory refresh controller associated with the first processor and configured to control refresh operations of the memory;

20 a second local memory refresh controller associated with the second processor and configured to control refresh operations of the memory; and

a refresh line connected to the first local memory refresh controller and the second local memory refresh controller over which refresh signals can be received by both the first local memory refresh controller and the second local memory refresh controller.

25

10. The apparatus of claim 9, wherein the first and second local memory refresh controllers are of common configuration.

11. The apparatus of claim 10, wherein the first and second local memory refresh
30 controllers operate synchronously.

12. The apparatus of claim 9, wherein each of the first and second local memory refresh controllers comprises a dedicated input pin, a refresh counter and a reset.

13. The apparatus of claim 9, wherein the memory is a DRAM, and wherein the first
5 and second local memory refresh controllers are local DRAM refresh controller.

14. A method of arbitrating mastership of a bus, comprising:
providing a first processor having a first local arbitrator and a second processor
having a second local arbitrator of common configuration to the first local arbitrator;
10 asserting, a first signal indicative of a first request for mastership of the bus on a first
bus request line associated with the first processor;
asserting, a second signal indicative of a second request for mastership of the bus on a
second bus request line associated with the second processor;
sensing the first and second signals by the first and second local arbitrators;
15 determining mastership of the bus by the first local arbitrator;
determining mastership of the bus by the second local arbitrator; and
granting mastership of the bus to one of the first and second processors without
requiring results of determined mastership to be communicated between the processors.

20 15. The method of claim 14, wherein mastership is determined simultaneously by the
first local arbitrator and the second local arbitrator.

16. The method of claim 14, further comprising:
asserting, by a first processor, a signal indicative of a priority request;
25 sensing, by the second arbitrator, the signal indicative of the priority request; and
de-asserting, by the second processor, the second signal.

17. A method, comprising:
asserting over a first bus request line a first request for mastership of a bus by a first
processor;
30 asserting over a second bus request line a second request for mastership of the bus by
a second processor;

- monitoring by a first local arbitrator associated with the first processor the first and second bus request lines;
- monitoring by a second local arbitrator associated with the second processor the first and second bus request lines;
- 5 determining by the first arbitrator that the second processor is to become the next bus master;
- determining by the second arbitrator that the second processor is to become the next bus master; and
- granting mastership of the bus to the second processor without communicating to the
- 10 first processor that the first processor will not become the bus master.

FIG. 1

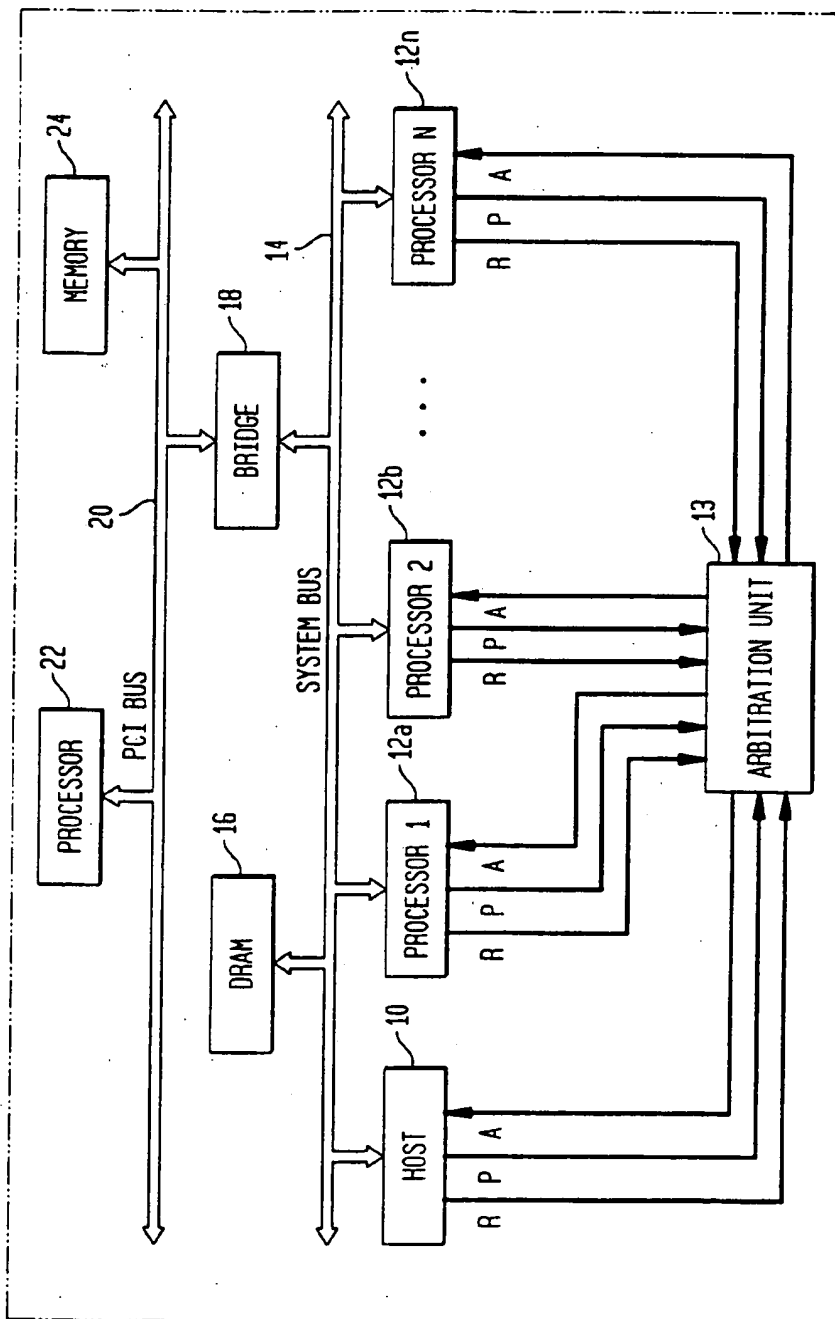


FIG. 2

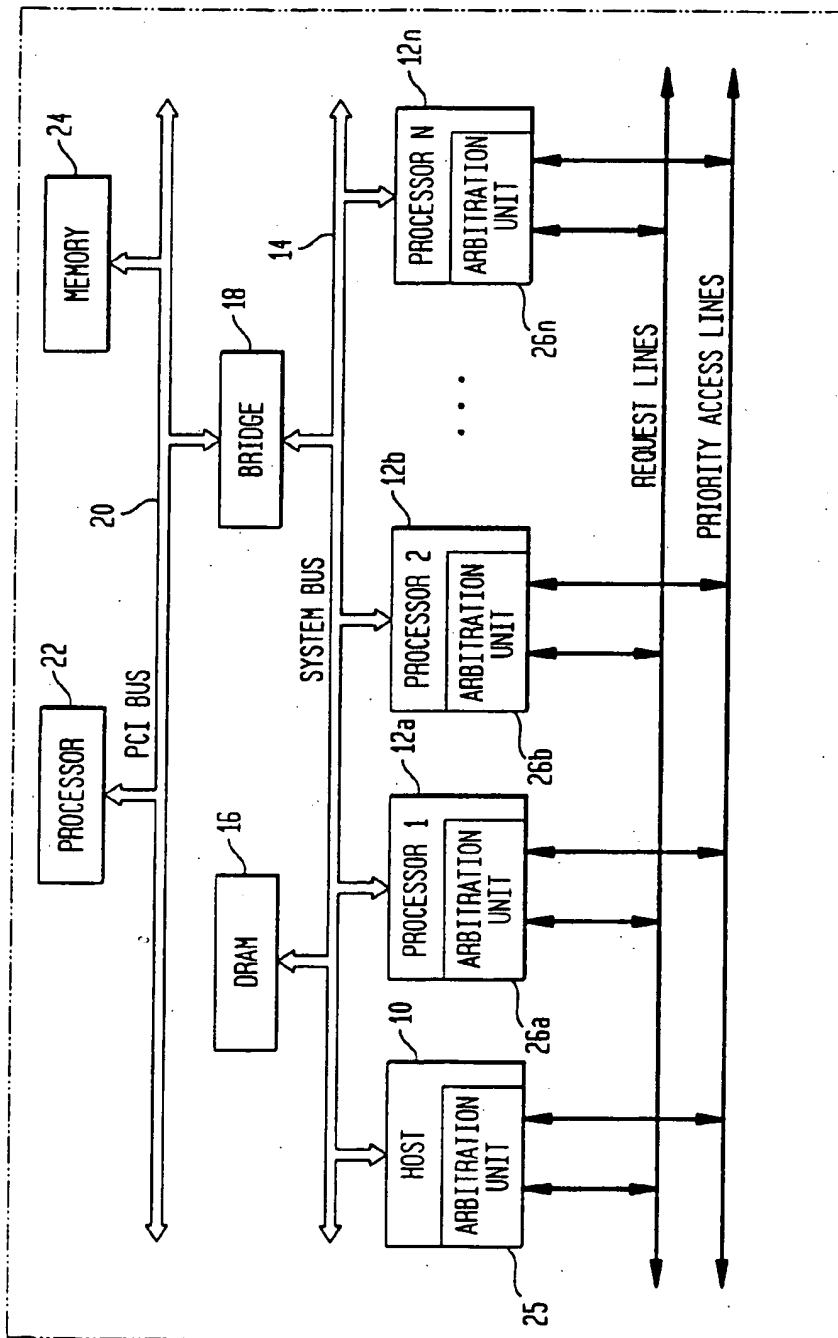
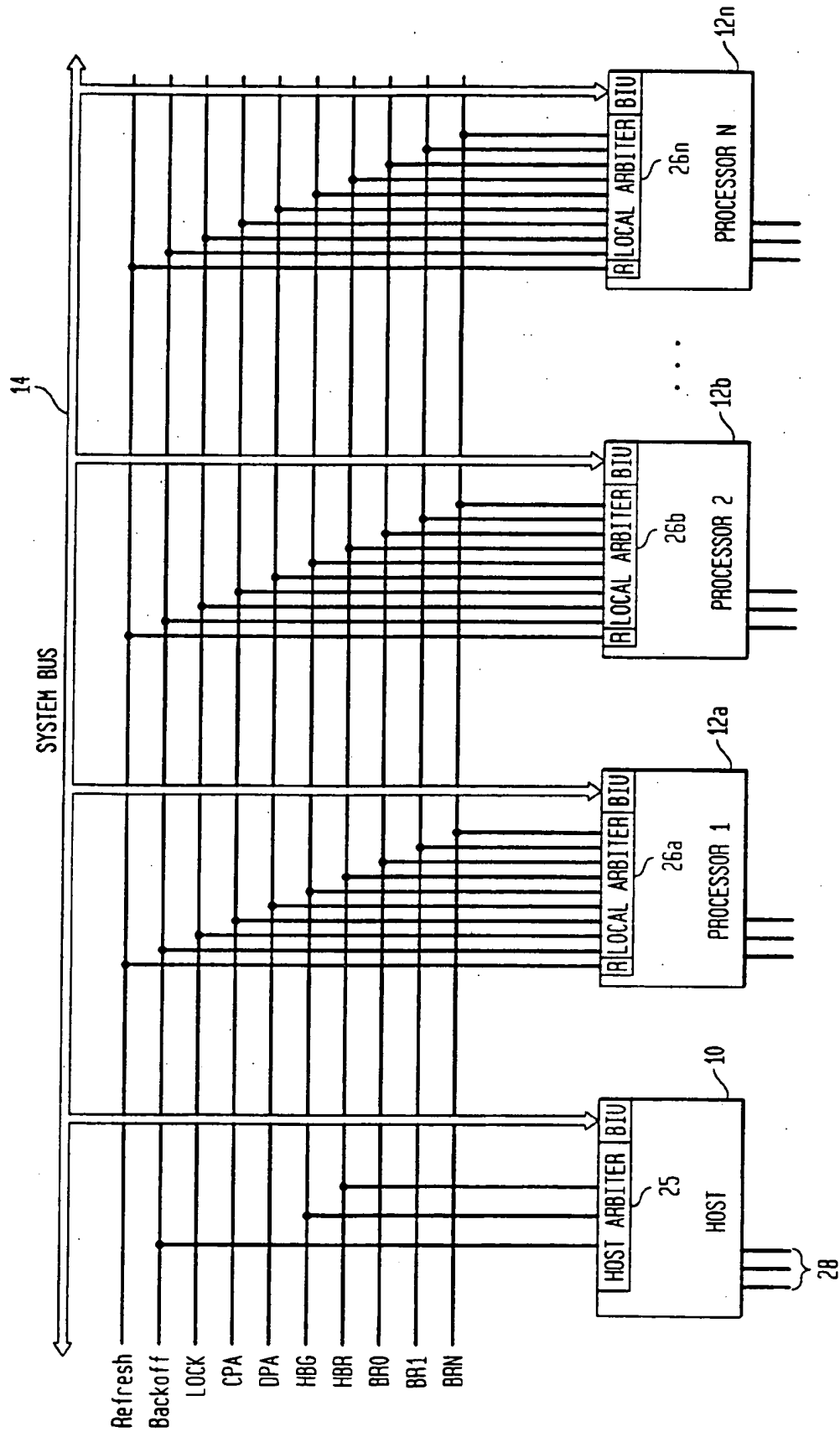
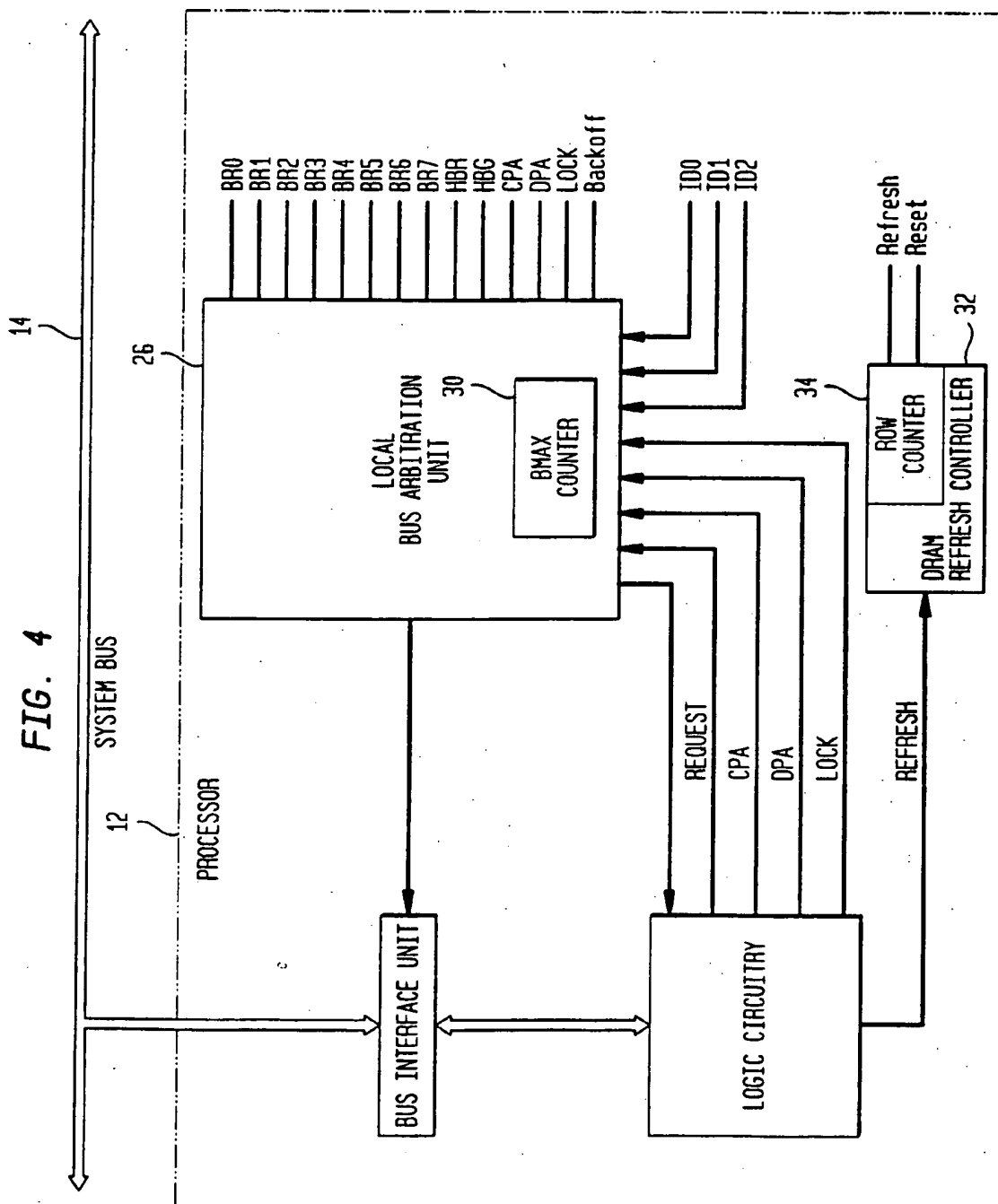


FIG. 3





5/7

FIG. 5

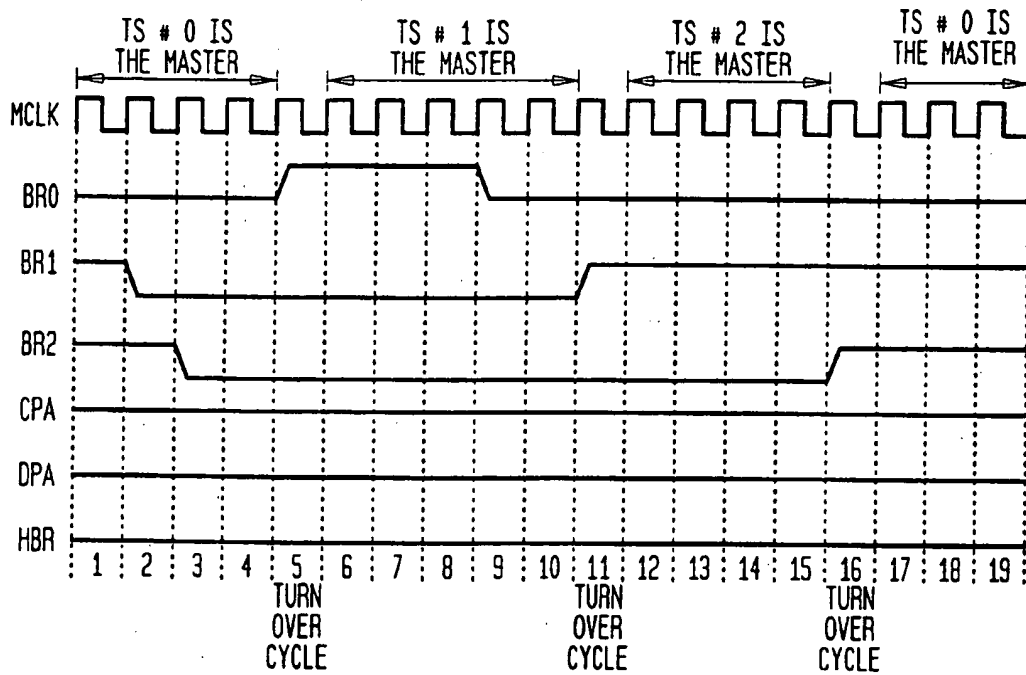
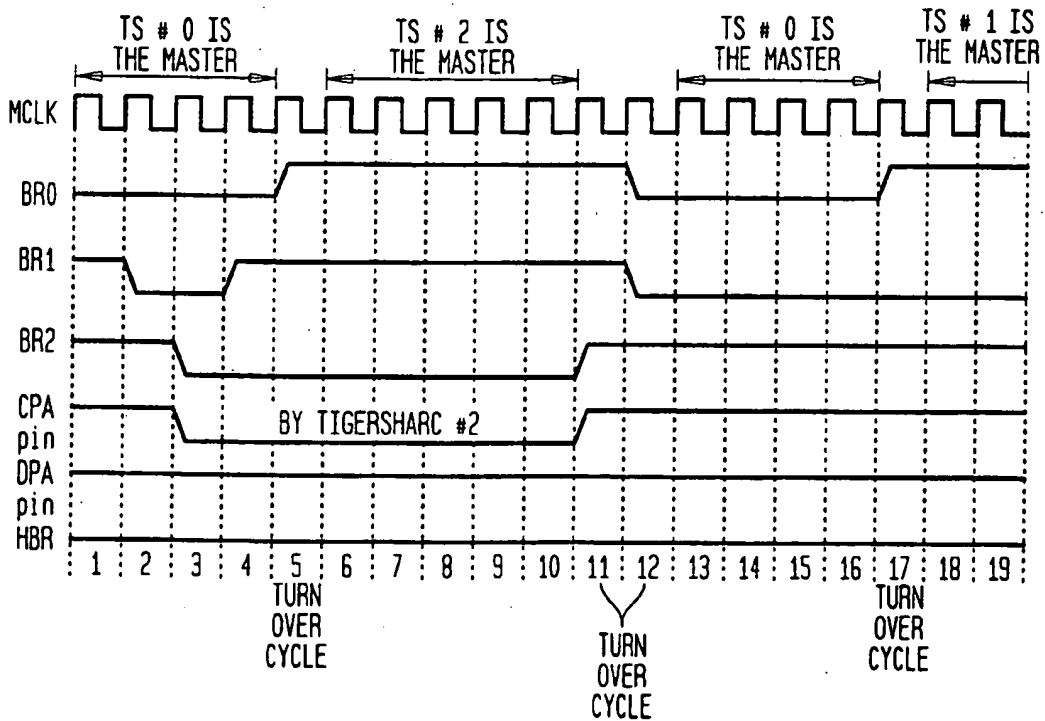


FIG. 6



6/7

FIG. 7

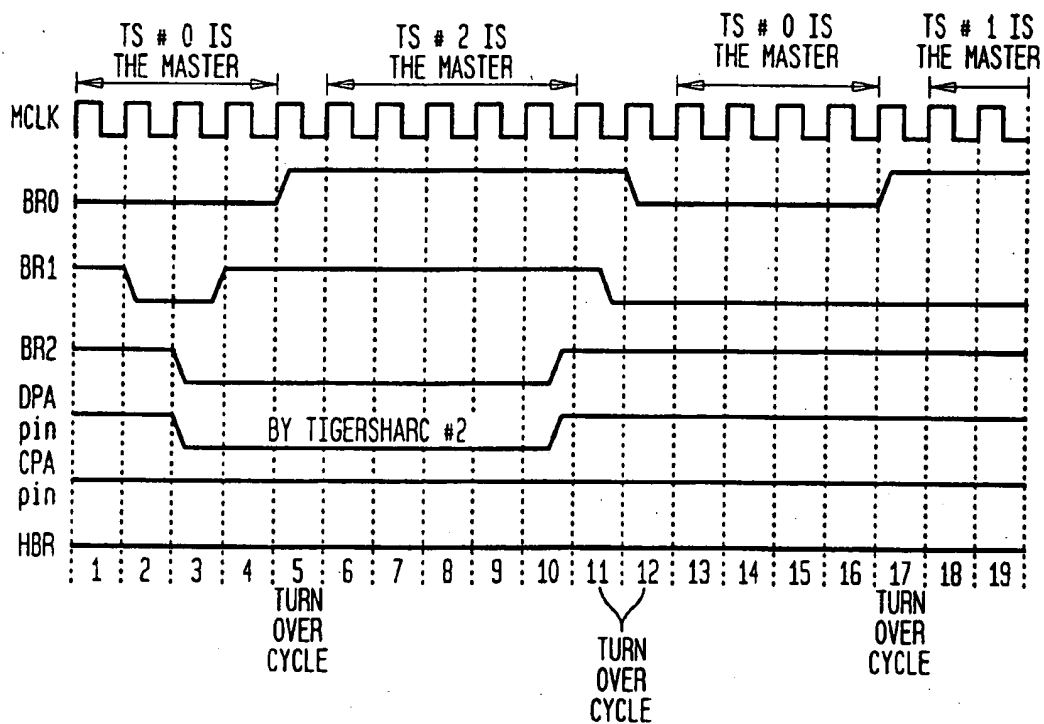
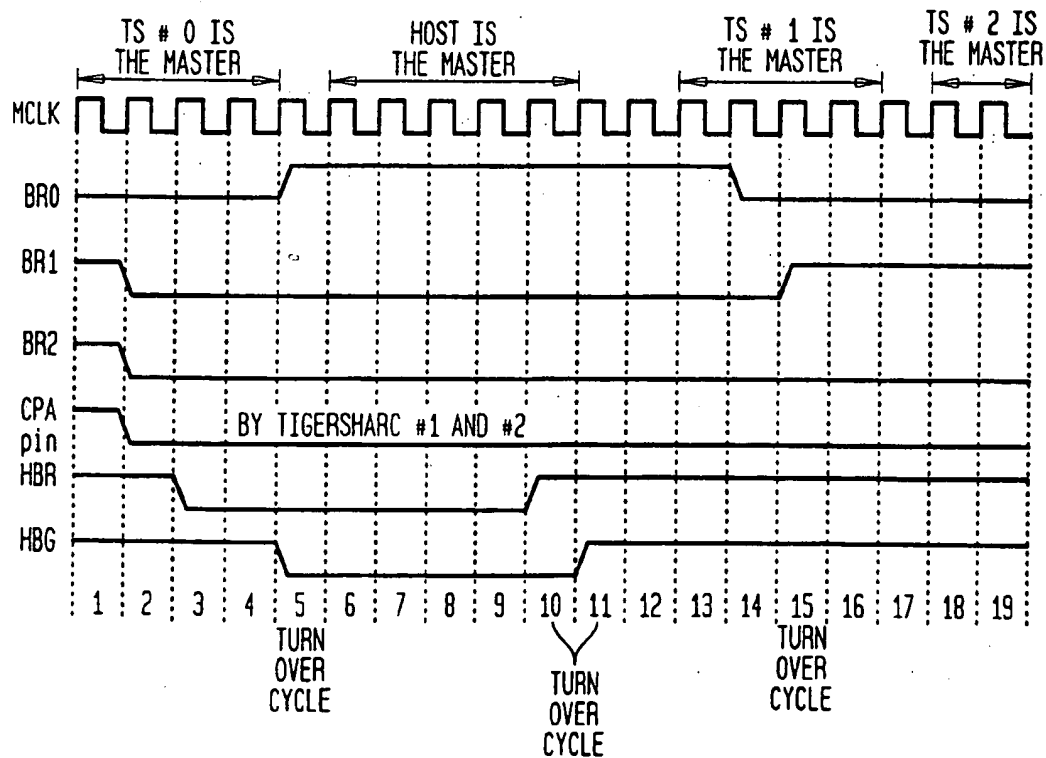
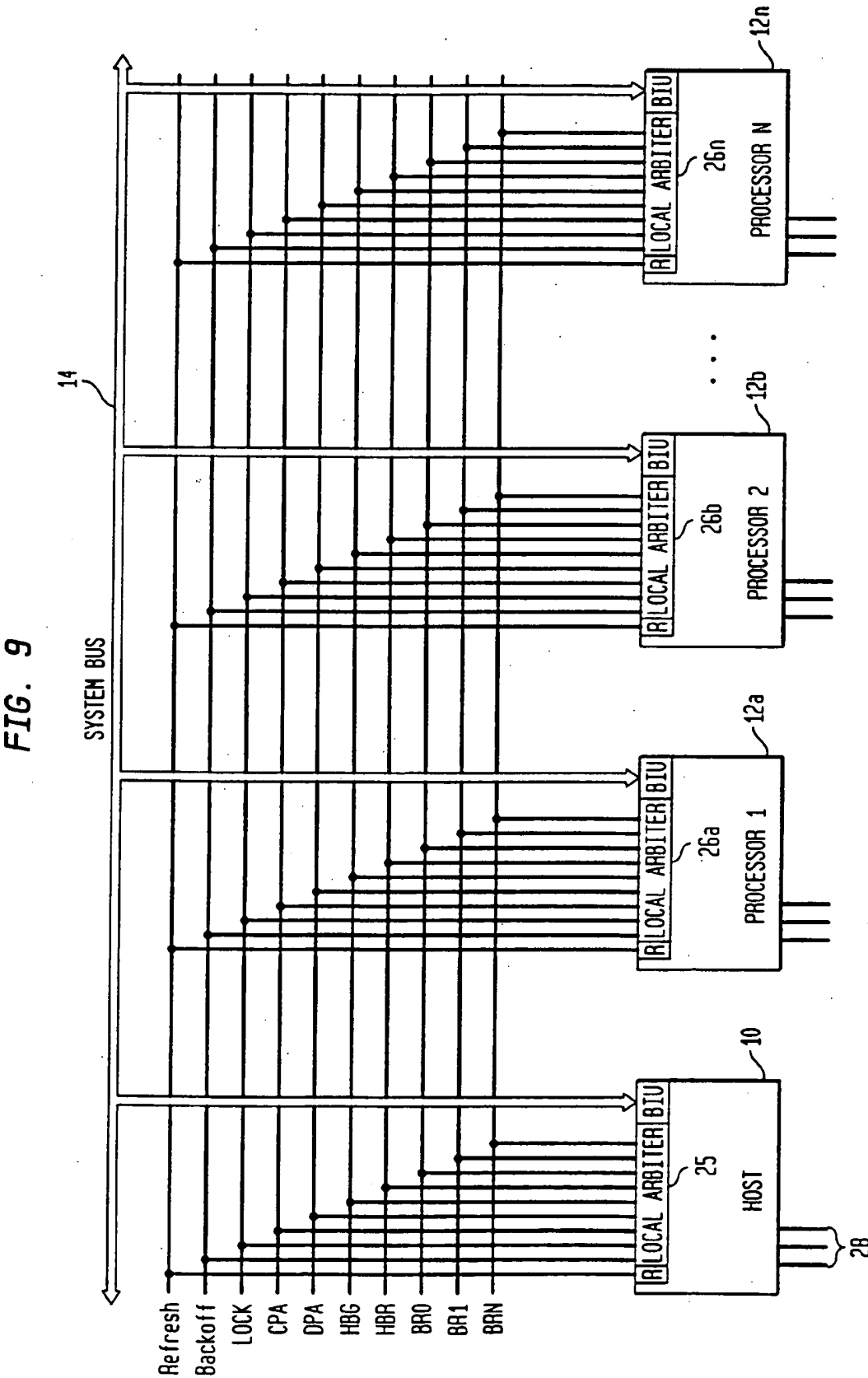


FIG. 8





INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 00/00951

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F13/368 G11C11/406

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F G11C

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	EP 0 737 925 A (SUN MICROSYSTEMS INCORPORATED) 16 October 1996 (1996-10-16) column 6, line 18 -column 9, line 43 figure 2	1-6, 14, 15, 17 6, 7, 16
X	US 5 313 591 A (AVERILL) 17 May 1994 (1994-05-17) column 3, line 16 -column 4, line 56 column 8, line 12 - line 26 figures 1, 2	1-5, 14, 15, 17
X	US 5 261 109 A (CADAMBI ET AL) 9 November 1993 (1993-11-09) column 2, line 1 -column 3, line 14 column 9, line 27 -column 10, line 8 -/--	1-5, 14, 15, 17

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

9 June 2000

Date of mailing of the international search report

19/06/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 eponrl,
Fax: (+31-70) 340-3016

Authorized officer

McDonagh, F

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 00/00951

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5 265 223 A (BROCKMANN ET AL) 23 November 1993 (1993-11-23) column 2, line 40 -column 4, line 26 figures 1,2 ----	1-5,17
Y	US 5 301 283 A (THACKER ET AL) 5 April 1994 (1994-04-05) column 3, line 59 -column 4, line 62 column 6, line 65 -column 9, line 3 ----	6,7,16
A	US 4 796 232 A (HOUSE) 3 January 1989 (1989-01-03) column 7, line 36 -column 8, line 14 column 14, line 9 -column 15, line 27 figure 1 ----	9-13
A	US 5 638 529 A (YEE ET AL) 10 June 1997 (1997-06-10) column 2, line 50 -column 3, line 50 claims 1-5; figures 3,4 -----	9-13

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 00/00951

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 737925 A	16-10-1996	US 5710891 A CA 2171170 A JP 9026933 A US 5862356 A	20-01-1998 01-10-1996 28-01-1997 19-01-1999
US 5313591 A	17-05-1994	NONE	
US 5261109 A	09-11-1993	NONE	
US 5265223 A	23-11-1993	NONE	
US 5301283 A	05-04-1994	NONE	
US 4796232 A	03-01-1989	NONE	
US 5638529 A	10-06-1997	NONE	